1D wave equation with finite elements

INF5620

2014

Project 1: 1D wave equation with finite elements

The purpose of this project is to derive and analyze a finite element method for the 1D wave equation

$$u_{tt} = c^2 u_{xx}, \quad x \in [0, L],\ t \in (0, T],$$

with boundary and initial conditions

$$u(0,t) = U_0(t), \quad u_x(L,t) = 0, \quad u(x,0) = I(x), \quad u_t(x,0) = V(x)\,.$$

The analysis will give insight into how to adjust the default behavior of the finite element method such that its properties are as good as those for the finite difference method for this particular equation. With the necessary adjustments discovered in detailed 1D calculations, one gets a recipe for constructing an accurate finite element method for simulating 2D and 3D waves in complex geometries.

Relevant background material consists of

- Time-dependent finite element discretization[1], which builds on finite element discretization in space[2].
- Analysis of wave equations[3], which builds on analysis of vibration equations[4].

Introduce a set of $N_n = N - 1$ nodes numbered from left to right, with coordinates $x_0, x_1, \ldots, x_N$. Associated with these nodes are a set of basis functions $\{\varphi_0(x), \ldots, \varphi_N(x)\}$.

Let $u_{\mathrm{e}}(x,t)$ be the exact solution of the initial-boundary value problem. After discretization in time by finite differences, we get a set of time-discrete

[1] `http://tinyurl.com/k3sdbuv/pub/sphinx-fem/._main_fem019.html`
[2] `http://tinyurl.com/k3sdbuv/pub/sphinx-fem/._main_fem013.html`
[3] `http://tinyurl.com/k3sdbuv/pub/sphinx-wave/._main_wave004.html`
[4] `http://tinyurl.com/k3sdbuv/pub/pub/sphinx-vib/._main_vib000.html#analysis-of-the-numerical-scheme`

problems for $u_{\mathrm{e}}^n(x)$, where $u_{\mathrm{e}}^n(x) \approx u_{\mathrm{e}}(x, t_n)$. These functions of space are then approximated by finite element expansions

$$u_{\mathrm{e}}^n(x) \approx u^n(x) = \sum_{j \in \mathcal{I}_s} c_j^n \varphi_j(x), \tag{1}$$

in case the linear system for the coefficients $\{c_j^n\}_{j \in \mathcal{I}_s}$ is modified such that $c_0^j = U_0(t_n)$. The unknown coefficients now involve u at all the nodes and consequently $\mathcal{I}_s = \{0, \ldots, N\}$.

Alternatively, one can use a boundary function to take care of the Dirichlet condition at $x = 0$:

$$u^n(x) \equiv u(x, t_n) = U_0(t_n)\varphi_0(x) + \sum_{j \in \mathcal{I}_s} c_j^n \varphi_{j+1}(x)\,. \tag{2}$$

Now, $\mathcal{I}_s = \{0, \ldots, N-1\}$ and the unknowns $c_0^n, \ldots, c_{N-1}^n$ are the values of u at the N nodes $x_1, \ldots, x_N$, i.e., all the nodes where we do not have Dirichlet conditions.

a) Set up equations for the coefficients c_j^0, $j \in \mathcal{I}_s$, associated with the initial condition $u(x, 0) = I(x)$. Use two methods: Galerkin and collocation/interpolation.

b) Use a finite difference method in time to formulate a sequence of spatial problems for $u_{\mathrm{e}}^n(x)$, $n = 0, 1, \ldots, N_t$. A special problem is needed for $n = 1$ in order to incorporate the boundary condition with $u_t(x, t)$.

c) Apply Galerkin's method to derive a variational form of each of the spatial problems. Integrate the term with the second-order derivative by parts. Express the variational form in terms of u^{n+1}, u^n, and u^{n-1}.

d) Insert (1) or (2) in the variational form and derive the linear system to be solved at each time level. Express the system in the form

$$Mc^{n+1} = 2Mc^n - Mc^{n-1} - \Delta t^2 c^2 K c^n,$$

where M and K are matrices, $c^n = \{c_j^n\}_{j \in \mathcal{I}_s}$. Set up the formulas for the matrix entries $M_{i,j}$ and $K_{i,j}$.

Remark. Unless M is diagonal, a (tridiagonal) *linear system* must be solved at each time step, whereas the finite difference method leads to an explicit formula for u_i^{n+1} at each spatial point at a new time level.

e) Use P1 elements and compute element matrices and vectors corresponding to the M and K matrices. Assemble the element contributions to a global matrices.

f) Interpret equation number i in the linear system as a finite difference approximation of $u_{tt} = c^2 u_{xx}$ using the following scheme:

$$[D_t D_t (u + \frac{1}{6}\Delta x^2 D_x D_x u) = c^2 D_x D_x u]_i^n\,. \tag{3}$$

Hint. Write out an arbitrary equation in the linear system and group the unknown coefficients to mimic the differences above. Then substitute the coefficients by their corresponding u values, using a notation u_i^n as the finite element approximation of $u_{\mathrm{e}}(x_i, t_n)$, and write the finite element equation in the same form as a finite difference scheme.

g) Perform an analysis of the scheme (3) in the same way as is done for the corresponding finite difference scheme in the course notes[5]. That is, investigate a Fourier component $u_p^n = \exp\left(i(kp\Delta x - \tilde{\omega} n\Delta t)\right)$. Show that the stability criterion is

$$C \leq \frac{1}{\sqrt{3}},$$

and therefore that any hope for recovering the exact solution for $C = 1$ becomes impossible.

h) Find the numerical dispersion relation ($\tilde{\omega}$ expressed by other parameters) and plot the error in wave velocity $\tilde{c}/c$ ($\tilde{c} = \tilde{\omega}/k$, $c = \omega/c$) as a function of $k\Delta x$ for various Courant numbers. Compare with the corresponding plot for the finite difference method[6] for $u_{tt} = c^2 u_{xx}$ (computed with the aid of the program `dispersion_relation_1D.py`[7]).

i) Use the Trapezoidal rule to compute the M matrix. Show that the finite element method with P1 elements now produces the same scheme at the interior mesh points as the standard finite difference method for $u_{tt} = c^2 u_{xx}$. Also show that the first and last equations, which are affected by the boundary conditions, also are identical in the two methods.

j) Instead of using the Trapezoidal rule to produce a diagonal M matrix, one can replace M by $\mathrm{diag}(Me)$, where $e = (1, 1, \ldots, 1)$ and $\mathrm{diag}(w)$ means a diagonal matrix with the vector w on the diagonal. The interpretation of this approach is that we sum each row in M and place the sum on the diagonal. Show that this method produces the same results as Trapezoidal integration (in a 1D problem). Also show that if you replace each row in the element matrices associated with M by the row sum on the diagonal, the same result arises.
Filenames: `wave1D_fem.py`, `wave1D_fem.pdf`.

Remarks. Say we want to solve the 3D wave equation $u_{tt} = c^2\nabla^2 u$ with finite elements and get the same stability as in the finite difference method. We can then compute M and K in the usual way and thereafter just replace M by $\mathrm{diag}(Me)$. When the mass matrix is lumped this way and becomes diagonal, we get the same representation of of the terms u^{n-1}, u^n, and u^{n+1} as in the finite difference method. The K matrix also resembles finite differences for P1 elements. With Q1 elements the K matrix has more nonzero entries per row, but this does not change the numerical properties of the scheme significantly. The key issue is to use a lumped mass matrix.

[5] `http://hplgit.github.com/INF5620/doc/pub/sphinx-wave/main_wave.html#analysis-of-the-continuous-and-discrete-solution`
[6] `http://hplgit.github.com/INF5620/doc/pub/sphinx-wave/main_wave.html#wave-pde1-fig-disprel`
[7] `https://github.com/hplgit/INF5620/blob/gh-pages/src/wave/dispersion_relation_1D.py`